APPENDIX F:
USING THE COMPACT DISC

CD-INFO Information for Users of the ALE Clean Tones Compact Disc 01a

TABLE OF CONTENTS

1. Protocol Files for Generating ALE Calls

Protocol files are ASCII text files that describe the particular type of call and data that are used by a FED-STD 1045A/MIL-STD 188-141A HF ALE controller. The protocol file is a simple representation of the call, understandable by those who use the ALE equipment, but also used by the computer software to generate the actual call. The protocol is read by the ALECALL software program which generates an ASCII tone sequence file and a digitized binary sound file to be placed on a Compact Disc or played through a high quality sound card. The sound file is produced in one of several formats: the default is a Microsoft Wave file for use with a sound card, or the PCM or DA digital audio file is created for recording onto audio compact discs. An example of a simple call in which JOE calls SAM to establish a link is shown.

```
# ALE Protocol file for 1045 Radios
# Developed by Dave Wortendyke at NTIA/ITS.N1 on 10-26-92
# This is a comment line
# 1st character in each line defines type of line, ie
#     a '#' defines a comment line
#     a '*' defines a time delay or pause in integer Trw's (n*0.392 s)
#     a '!' defines total silence in integer Trw's (n * 0.392 s)
#     an 'L' means the Left or Originate radio tones
#     an 'R' means the Right or Called radio tones
# 2nd word is either one of the 8 FED-STD 1045 preamble codes/commands
#     or a '*' for a tick mark for net calls
# 3rd word is the information/data word of exactly 3 characters
#     or a fraction like 14/3 Trw's representing spacing of net slots
# 4th word is a number telling how many times the line will be used
# The scanning call duration in this test is based upon the UUT set to
#     scan 10 channels at 2 channels/sec., for a total of 5 seconds.
#
L     TO    SAM   15
L     TIS   JOE   1
*     2
R     TO    JOE   2
R     TIS   SAM   1
*     2
L     TO    SAM   2
L     TIS   JOE   1
```

A time delay is represented in the simple example above by a "* 2". This equals two T_{rw}s or 2 * .392 seconds. Later, when one advances to more complex calls, such as net and group calls, the protocol for time delays is extended to include not only the T_{rw} (392 ms) but the T_w timing (392/3 ms). The format of a fractional T_{rw} becomes

```
*     L     2/3         # Time delay of 2 Tw = 2/3 Trw
```

A more realistic example where the two protocol files are different, might be when JOE calls SAMMY and sends an AMD in the acknowledgment.

```
#   Protocol File -----------
L      TO       SAM      14
L      DATA     MY@      1
L      TO       SAM      1
L      DATA     MY@      1
L      TIS      JOE      1
*      2
R      TO       JOE      2
R      TIS      SAM      1
R      DATA     MY@      1
*      2
L      TO       SAM      1
L      DATA     MY@      1
L      TO       SAM      1
L      DATA     MY@      1
L      COMMAND  HEL      1
L      DATA     LO^`     1
L      REPEAT   THE      1
L      DATA     RE^`     1
L      TIS      JOE      1
```

In the above protocol file, the two-ALE-word, 5-character "HELLO" must be padded by a blank character. Since the blank character (ASCII 32) is a nonprintable character in the ASCII set, a different method of representing the space (and other nonprintable characters) had to be found. This technique is described in Section 2 which follows.

2. ASCII Text Protocol Files Representing All 128 Characters

The purpose of creating an ASCII text protocol file is to provide the user of a software program that controls various hardware functions with a file that can be changed with a simple text editor, and then printed on a standard printer using only the 95 printable ASCII characters. Normally the protocol file should contain only the standard 95 printable characters, but sometimes it is necessary to use control characters, or the full 128 ASCII set. The method below describes a technique to represent all 128 characters with the 95 printable ones.

The ground rules are as follows.

> All the **printable** characters will be printed as single characters **as normal** except for the backslash and the up-arrow. The backslash "\" (ASCII 92) and up-arrow "^" (ASCII 94) will be printed as doublets using the backslash as the leading character. The DEL (ASCII 127) will be represented by the doublet backslash and the printable character "@".
>
> The normal control character set and space will be printed using a doublet sequence with the up arrow (ASCII 94) as the lead character, and the printable ASCII character whose value is 64 more than the control character.

Examples:

Character	**ASCII decimal**	**Printed**	**Doublet ASCII Values**
NUL	0	^@	94, 64
SOH	1	^A	94, 65
STX	2	^B	94, 66
:	:	:	
RS	30	^^	94, 94
US	31	^_	94, 95
space	32	^`	94, 96
!	33	!	n/a
:	:	:	
Z	90	Z	n/a
[	91	[	n/a
\	92	\\	92, 92
]	93	]	n/a
^	94	\^	92, 94
_	95	_	n/a
:	:	:	
}	125	}	n/a
~	126	~	n/a
DEL	127	\@	92, 64

3. Creating a Protocol File for an ALE Encoding (FED-STD lO45A)

For simple calls one merely follows the rules provided as comments in the sample protocol files. It is necessary to use all upper case characters for the eight types of preambles. One may use the entire word, but only the first three characters are significant. They are as follows.

Type	**Word Type**	**Code**
DAT	DATA	0
THR	THRU	1
TO	TO	2
COM	COMMAND	3
FRO	FROM	4
TIS	THIS_IS	5
TWA	THIS_WAS, TWAS	6
REP	REPEAT	7

The address or data word is always three contiguous characters that are case sensitive. In some situations where a control or special character is used, a doublet character is required to represent the single character. (All control or special characters start with "\" or "^".)

Many of the powerful features of the ALE coding utilize information at the bit level, which makes the generation of the protocol file difficult if done manually. The package includes a series of software modules or programs to automatically provide the translation. The various command functions and preprocessors are named as follows.

Type	**Cmd Code letter**	**Routine Name**
AMD	n/a	c-amd
LQA	a	c-lqa
DTM	d	c-dtm (Basic DTM only, not extended)

An example for the LQA preprocessor is provided. The desired function is a scanning call with LQA information:

Left COMmand a-C-M-S-B

where lower case 'a' denotes an "Analysis," i.e., LQA function,
the C is a control bit, 1 = return LQA, 0 = no return LQA,
the M is an octal value for Multipath (0-7),
the S is a 5 bit number for SINAD (0-9, A-V or other method for 0-31),
the B is a 5 bit number for BER (0-9, A-V or other method for 0-31).

The output of the "c-lqa" module is shown.

```
input:     Left    COMmand     a-1-7-K-5   (note: K=21= 10101, 5= 00101)

                         a    1  7    K      5              a      }       %
           word= 110 1100001 1 111 10101 00101 => 110 1100001 1111101 0100101

output:    LEFT    COMMAND     a}%
```

If we change the request for LQA to suppress the return LQA then we get:

```
input:     Left    COMmand     a-0-7-K-5   (note: K= 10101, 5= 00101)

                        a     0  7    K      5              a       =       %
           word= 110 1100001 0 111 10101 00101 => 110 1100001 0111101 0100101

output:    LEFT    COMMAND     a=%   (note the '}' became '=')
```

4. Tone Files

These files describe the audio tones that will modulate a high frequency radio signal transmission. They carry address and data information in a precise format specified by FED-STD-1045.

The tone files are produced as an intermediate step in the generation of the audio sound files by the ALECALL program. Basically, the data in the file specifies what "goes over the air," in which direction, and when. The first word of each line of the ASCII tone file contains the encoded data. It normally consists of octal digits, or a "*". A "#" anywhere in the line denotes comments or information that is not really part of the tone database. ASCII characters "T" and "$" are also allowed. The octal digits (0,1,2..7) and the "*" each represent one ALE baud or 8 ms of information. They are defined as:

Character	Meaning	Length (Cycles)	Time (ms)
0	Cosine wave of 750 Hz	6.000	8.000
1	Cosine wave of 1000 Hz	8.000	8.000
2	Cosine wave of 1500 Hz	12.000	8.000
3	Cosine wave of 1250 Hz	10.000	8.000
4	Cosine wave of 2500 Hz	20.000	8.000
5	Cosine wave of 2250 Hz	18.000	8.000
6	Cosine wave of 1750 Hz	14.000	8.000
7	Cosine wave of 2000 Hz	16.000	8.000
*	Silence of 8 ms	n/a	8.000

The tick mark "T", is used on the right stereo channel for all calls that require a response in a time slot. It is either a 4-ms cosine wave of 1000 Hz followed by silence, or a chime and a verbal indication of the number position of the current slot followed by silence. For example, at the start of the fifteenth slot, one would hear a chime and the word "fifteen." The chime is used on a few calls with 16 response slots. The duration of the tick (or chime) and silence is determined by the word time (T_w) given in the protocol control file and never exceeds the slotwidth. If the total redundant word time ($T_{rw} = 3 * T_w$) is not an integer for the slots described in the record of the protocol control file, the silence marks ("*") end with a "$" as a warning flag to an incomplete ALE word boundary.

The tone file described above may include additional information preceded by a "#". Each of the fields or following words must have "white space" as separators between them. The first word after the "#" is the stereo channel: LEFT, RIGHT, or BOTH. The next word is "time=" and the last word represents the total elapsed time in seconds since the start of the ALE call described in the protocol file.

The ALECALL program (command line option -t FILENAME) reads tone files (*.TON) to produce a sound file that could be used for calibration and to check levels and frequencies. The calibration tone file must follow the format of the ALECALL tone file output, except that there can be any number of tones per line on the calibration tone file (rather than exactly 49 as produced by ALECALL.)

5. Output Level

The default output level is currently set at 0.5 * 2^16 bits; this is 6 dB below the maximum voltage. This value represents 15 bits of the A-D converter. The generation software, ALECALL, varies this level at file generation time if so specified by the user. Call 12A on CD-01A has half of the default level (0.25 * 2^16) another 6 dB below the output level of the remainder of the CD. This was necessary due to the wide excursions of amplitude of the output that has Gaussian noise added.

6. Test Set Up

We have used with equal success the output of both very inexpensive and very good quality compact disc players (price range: $125 - 650) from the variable head phone output to drive the VOX input of a transceiver. The audio output of a low quality CD player is noticeably inferior to the human ear, but an eight digit frequency counter gives the same frequency and stability for both players. There is a wide variance in output frequency for some brands of CD players. Both the CD players we use have better than 10 ppm accuracy when the 1000.000-Hz tone is played (i.e. 1000.000 ± Δf, where $\Delta f < 0.010$ Hz). The 32-Ohm output of the CD player is coupled via an audio transformer to the 600-Ohm balanced VOX audio input on the transceiver. The transceiver is set to one of the channels which the ALE UUT is scanning. The two radios are coupled with two 30-dB power attenuators and a small pad usually set at 40 dB. Our experience has shown it is very important to:

1. First transformer couple or use a passive bridge to match impedances between the CD phone output and the transceiver.
2. Then adjust the CD player output level and the transceiver VOX gain to a proper level, monitoring the SINAD or channel score until the maximum is reached.

Failure to perform the above steps may cause distortion that will affect the performance testing portion of the test plan (degraded tones CDs.) We were able to drop the LQA scores from a maximum of 30 to 18-22 by setting the output level either too low or too high on the CD player.

7. Disc Specifics - Audio Compact Disc 01a

The first track on CD-01a was recorded in an on-site laboratory with a microphone hooked directly to a computer disk. It is meant to be an informative introduction, and the quality of the track is of no concern, since the rest of the disc was produced totally from computer software and has the best S/N ratio that a CD player can produce. All test data (ALE tones) are on the left channel. Some new net and group calls on the final CD-01a disc have a few responses in the net response slots which are also on the left channel. The right channel is reserved for information and tick marks (for net calls, group calls, etc.). These simulate other stations which are responding to the net call. The CD track layout sheet marks those tests that are optional in FED-STD-1045 with an "*". Those tests are required for FED-STD-1046/1. Tests numbered 13 and higher are marked with an "#" and are optional because ITS included them without verification against the Federal Standard.

8. Why an Audio CD Was Chosen over a CD-ROM

The audio CD is identical to the music CDs sold in stores for $5 - $15 each that are played on a home audio CD stereo player. Audio CDs are now so popular that there are portable CD players and automobile CD players. The disc can hold 60 to 70 minutes of audio recording (i.e., stereo music, voice, or sound tracks.) Using a calibrated square wave on a test/demonstration audio CD or our own 1000-Hz cosine wave calibration track two, the accuracy of several home CD players varied from 7 parts per million (ppm) to 108 ppm when measured with an 8-digit frequency counter. Data for the audio tracks is transferred to an ISO optical compact disc writer from the computer disk drive. The writer must run nonstop, which records at a rate of approximately 10 Mbytes/minute. The prototype CD thus produced is sent to a CD manufacturer after testing in the laboratory. A master CD and submasters are made from the prototype, so the copies may be pressed out from a reverse image submaster.

The other CD popular format is the CD-ROM, which is more like an extremely high capacity floppy disk. (Note the spelling: disks are magnetic media, discs are optical media.) There are several formats that could be used for the CD-ROM, including ISO 9660 and Rockridge formats. Most CD-ROMs have the dual function of being able to be used as a slow disk drive for programs and data files, and with special software as a playback for audio CDs. The CD-ROM is not intended for use as an audio CD player, but will work. Note an accuracy of 84 ppm was measured when an audio CD was played in a good CD-ROM unit. By using a high quality DSP sound card (price = $380) a very good playback frequency accuracy can be obtained from 80486 computers. We measured 15 ppm accuracy using a 486/33 computer. Use of a lesser computer is not recommended. Thus, a disadvantage of choosing the CD-ROM over the audio CD format is that the equipment requirements for the user would be greater than those for the audio CD. An added advantage of the CD-ROM is that all the documentation and sample protocol files could be stored on the same CD with the digital sound files. Another disadvantage is the higher cost to produce a master CD. The cost would be about 80% more than the audio CD for the master and 200 - 300 discs. We decided against producing CD-ROMs due to the added expense of production and the additional equipment requirement of the user.

In summary, there are good reasons to favor each technology, the sound card, the CD-ROM, and audio CD. The biggest advantage of using an accurate sound card is that the users and testers can generate a new digital file (different than that provided on the CD) with a different set of station addresses for their unique application, using the ALECALL software provided on the floppy disk. The best reason for using an audio CD is that everyone who tests ALE radios will have the same set of tests used by others, and the disc reproduction is relatively inexpensive. For our development effort, we use the quality sound card, but for everyday testing the audio CD is very convenient and inexpensive. A combination obviously provides the best and most flexible test combination.

9. Executable Software Programs
9.1 ALECALL

ALECALL software program for generating ALE Clean Tones was released in January 1995. It was developed by NTIA/ITS in Boulder, Colorado. This version of the program will generate the ALE pure (clean) tones and tones with Gaussian noise.

At the request of the HFIA representatives, this program is being released to all organizations that request it or obtain it via the Internet. The software is not in final form, but in an initial release version with clean tones and tones with Gaussian noise. This software release has several purposes:

1. It provides a means of producing sound (Wave) files that may be played on a high quality sound card. We used the audio output from the sound card to test each protocol of the final issue of CD-01A before recording it to a CD. It is likely that the industry providers may quickly set up and verify the operation of their radios against the sound files. This would benefit both the Government and industry by identifying either mistakes in the CD, mistakes in the vendors' ALE DSP coding, or misunderstood sections of the Federal and military Standards for HF ALE radios. All of this can be accomplished in the privacy of each vendor's test laboratory.

2. It provides a sneak preview of the more sophisticated ALECALL software program to be released in the future. That final software release will include a full Watterson model software simulator to degrade the ALE modem tones. Noise, fading, and multipath (NFM) will be specified in the protocol files (*.PRO) and will include the full range of CCIR conditions to produce GOOD, MODERATE, and POOR channel conditions.

3. It provides the HF industry a chance to provide final comments on the content and organization of the Official ALE Clean Tone compact disc published in January 1995.

======================== **Using ALECALL program** ========================

```
Usage:        ALECALL ale-014a.pro          To invoke the software

Files made:   ALE-014a.ton                  The output tone (*.TON) file
              ALE-014a.wav                  The Wave sound file

--- OR--

Usage:        ALECALL -t ale-014a.ton       To invoke the software with a .TON file

Files made:   ALE-014a.wav                  The Wave sound file
```

====================== **ALECALL help screen** ==========================

```
               ALECALL -- NTIA/ITS, Boulder, CO
               Version 3.51; Jan, 1995 (tgs, gps)

A program to generate ALE tones from a Standard Usage: alecall [-arguments]
filename(s) arguments:
      ?,h   -> This help screen
      a     -> Write Digital Audio formatted output (.DA)
      t     -> Read the input as a .TON file
      c     -> Insert chimes instead of ticks
               (TICKS.DAT must be in directory)
      v     -> Verbose mode: Display extra status information
      q     -> Quick mode: Only generate .TON files

      Defaults:
            input files: ASCII Protocol File (*.PRO)
            output files: Wave file (.WAV)
      You must be in the same directory as the .pro files!
      Only Gaussian and Level NFM parameters in protocol files
            are accepted in this version
      Default level for this version is 0.50 (15 bits)
```

====================== **Hardware for ALECALL** =========================

Equipment required:
CPU 80486/33 (80386 and slower machines work, but take too long.)
Disk 10 to 40 MB free space
Sound card Good quality, with on board CPU and DSP chip
Warning: The *.WAV files accumulate at the rate of slightly over 10 Mbytes per minute of playing time. Check your free disk space.

9.2 ALETIME

ALETIME software program computes playing time from the protocol files. This program was essential for creating a CD track list. It is provided as an aid to test and disk space planning.

====================== **Using ALETIME program** ======================

Usage: `ALETIME ale-07?.pro` To invoke the software

Sample output:

```
Trk#  File Name     Playing   Total Trw's          Elapsed Time
                                 Time            seconds   min:sec
  1   ALE-07A.PRO   48.608       124.             48.608   0:48.608
  2   ALE-07B.PRO   50.960       130.             99.568   1:39.568
  3   ALE-07C.PRO   52.528       134.            152.096   2:32.096
  4   ALE-07D.PRO   23.912        61.            176.008   2:56.008
  5   ALE-07E.PRO   48.608       124.            224.616   3:44.616
  6   ALE-07F.PRO   27.440        70.            252.056   4:12.056
                            ____________      ______________________
  6   TOTALS                     643.00          252.056   4:12.056
```

Note: most CD recorders add 2 sec per track as a buffer, hence the calculated time shown above may be 12 sec less than the actual recorded time.

====================== **ALETIME help screen** ==========================

```
*******************************************************
*    ALETIME program to compute running times    *
*        of protocol (.pro) files for CD         *
*         U.S. Department of Commerce            *
*                NTIA/ITS.Nl                     *
*               December 1994                    *
*                Version 1.2                     *
*******************************************************

Use: ALETIME [- arguments] filename(s)
arguments:
      ?,h   : This help screen
      a     : To add 2 seconds per track
          (To compensate for most CD writers)
filename(s):
      One or more protocol (.pro) files
      (A wildcard may be used, i.e., *.pro)
```

9.3 C-AMD

C-AMD software program creates a protocol file with an AMD message. C-AMD facilitates building protocol files with AMD messages since the AMD command codes include nonprintable characters. (See Section 2 of this appendix for details on nonprintable characters or FED-STD-1045A for information on AMD messaging.)

========================== **C-AMD help screen** ==========================

```
***** C-AMD.C *****
Written by Chris Riddle, ITS.Nl
Version 1.0

Usage: c-amd <to> <from> <message.file>
   or: c-amd <to> <from> 'put message in single quotes'
      Output is ale-amd.pro which contains the correct protocol
      file for the given AMD message.
```

========================= **Using C-AMD program** ========================

Usage: `C-AMD BOB DOC 'this is a message'` To invoke the software

--- OR --

Usage: `C-AMD BOB DOC MESSAGE.TXT` To invoke the software

Sample input file: MESSAGE.TXT

```
Hello, this is a sample AMD for your review! "(12345*&%)
```

Sample output file: ALE-AMD.PRO

```
#  ALE Protocol file for 1045 Radios
#  Developed by Dave Wortendyke at NTIA/ITS.N1 on 10-26-92
#  This is a comment line
#  1st character in each line defines type of line, ie
#       a '#' defines a comment line
#       a '*' defines a time delay or pause in integer Trw's (n *0.392 s)
#       a '!' defines total silence in integer Trw's (n * 0.392 s)
#       an 'L' means the Left or Originate radio tones
#       an 'R' means the Right or Called radio tones
#  2nd word is one of the 8 FED-STD 1045 preamble codes/commands
#  3rd word is the information/data word of exactly 3 characters
#  4th word is a number telling how many times the line will be used
#  The scanning call duration in this test is based upon the UUT set to
#       scan 10 channels at 2 channels/second for a total of 5 seconds.
#
#  This file generated by c-amd.c
#  AMD message
#
#  to BOB
#  from DOC
#  the message is from the file message.txt
L     TO    BOB    16
L     TO    BOB    1
L     TIS   DOC    1
*     4
R     TO    DOC    1
R     TO    DOC    1
R     TIS   BOB    1
*     3
L     TO    BOB    1
L     TO    BOB    1
# the AMD message follows...
L     COMMAND       Hel   1
L     DATA          lo,   1
L     REPEAT        ^`th  1
L     DATA          is^`  1
L     REPEAT        is^`  1
L     DATA          a^`s  1
L     REPEAT        amp   1
L     DATA          le^`  1
L     REPEAT        AMD   1
L     DATA          ^`fo  1
L     REPEAT        r^`y  1
L     DATA          our   1
L     REPEAT        ^`re  1
L     DATA          vie   1
L     REPEAT        w^a^` 1
L     DATA          “ (1  1
L     REPEAT        234   1
L     DATA          5*&   1
L     REPEAT        %)^`  1
#
L     TIS   DOC    1
*     1
```

9.4 C-LQA

C-LQA software program creates a protocol file with an LQA. As with AMD, the nonprintable characters are used in the command LQA. (See Section 3 of this appendix and FED-STD-I045A for details on LQA encoding.)

============================ **C-LQA help screen** ========================

```
      ***** C-LQA.C *****
      Written by Chris Riddle, ITS.Nl
      Revised by Teresa Sparkman
      Version 2.1

      Usage: c-lqa <to> <from> <control bit> [[<SINAD> <BER>]| R] [Debug]
            Output is ale-lqa.pro which contains the correct protocol
            file for the given LQA message.
            R - for LQA request, SINAD & BER undefined
            control bit: 1 = return LQA, 0 = don't return LQA
            SINAD: 0 - 30
            BER: 0 - 30
      NOTE: The maximum BER or SINAD value is 30
```

=========================== **Using C-LQA program** =======================

Usage: `C-LQA  KAT DOC 1 R` To invoke the software

Sample output file: `ALE-LQA.PRO`

```
      # This file generated by c-lqa.c
      #
      # to KAT
      # from DOC
      # control bit = 1
      # LQA request
      # SINAD = N/A
      # BER = N/A
      # address the called station...
      L     TO    KAT   15
      L     TO    KAT   1
      L     COMMAND   a/@/@ 1
      L     TIS   DOC   1
      *     5
      R     TO    DOC   1
      R     TO    DOC   1
      R     TIS   KAT   1
      *     2
      L     TO    KAT   2
      L     TIS   DOC   1
      *     1
```

--- OR --

Usage: `C-LQA  KATHREEN DOC 1 30 30` To invoke the software

Sample output file: ALE-LQA.PRO

```
# This file generated by c-lqa.c
#
# to KATHREEN
# from DOC
# control bit = 1
# SINAD = 30
# BER = 30
# address the called station...
L     TO          KAT   15
L     DATA        HRE   1
L     REPEAT      EN@   1
L     TO          KAT   1
L     DATA        HRE   1
L     REPEAT      EN@   1
L     COMMAND     a/@/^ 1
L     TIS         DOC   1
*     5
R     TO          DOC   1
R     TO          DOC   1
R     TIS         KAT   1
R     DATA        HRE   1
R     REPEAT      EN@   1
*     2
L     TO          KAT   2
L     DATA        HRE   1
L     REPEAT      EN@   1
L     TIS         DOC   1
*     1
```

9.5 C-DTM

C-DTM software program creates a protocol file with a DTM message. As with LQA and AMD, this program facilitates the encoding of the DTM nonprintable command characters. (See FED-STD-I045A for details on DTM encoding.)

========================== C-DTM help screen ==========================

```
***** C-DTM.C *****
Written by Chris Riddle, ITS.Nl
Version 2.1

Usage: c-dtm <to> <from> <kd4> <message.file>
   or: c-dtm <to> <from> <kd4> 'put message in single quotes'
       kd4 = 1 -> ACK request, kd4 = 0 -> no ACK request

Output is ale-dtm.pro which contains the correct protocol
      file for the given DTM message.
```

========================= Using C-DTM program =========================

Usage: `C-DTM  JASON DOC 0   'this is a sample message'` To invoke the software

Sample output file: ALE-DTM.PRO

```
# This file generated by c-dtm.c
# The COMMAND DTM is in one long block
# This assumes the receiving station has DTM capability
#
# to JASON
# from DOC
# message file temp.msg
# kd4 = 0
L     TO    JAS   16
L     DATA  ON@   1
L     TO    JAS   1
L     DATA  ON@   1
L     TIS   DOC   1
*     4
R     TO    DOC   1
R     TO    DOC   1
R     TIS   JAS   1
R     DATA  ON@   1
*     3
L     TO    JAS   1
L     DATA  ON@   1
L     TO    JAS   1
L     DATA  ON@   1
# the DTM follows...
L     COMMAND     d6$   1
L     DATA        thi   1
L     REPEAT      s^`i  1
L     DATA        s^`a  1
L     REPEAT      ^`sa  1
L     DATA        mpl   1
L     REPEAT      e^`m  1
L     DATA        ess   1
L     REPEAT      age   1
L     DATA        ^M^J@ 1
L     COMMAND     z;i   1
L     TIS         DOC   1
*     1
```

--- OR ---

Usage: `C-DTM JASON DOC 1 TEST.MSG` To invoke the software

Sample input file: TEST.MSG

```
this is a test of orderwire
```

Sample output file: ALE-DTM.PRO

```
# This file generated by c-dtm.c
# The COMMAND DTM is in one long block
# This assumes the receiving station has DTM capability
#
# to JASON
# from DOC
# message file test.msg
# kd4 = 1
L     TO    JAS   16
L     DATA  ON@   1
L     TO    JAS   1
L     DATA  ON@   1
L     TIS   DOC   1
*     4
R     TO    DOC   1
R     TO    DOC   1
R     TIS   JAS   1
R     DATA  ON@   1
*     3
L     TO    JAS   1
L     DATA  ON@   1
L     TO    JAS   1
L     DATA  ON@   1
# the DTM follows...
L     COMMAND     dv'   1
L     DATA        thi   1
L     REPEAT      s^`i  1
L     DATA        s^`a  1
L     REPEAT      ^`te  1
L     DATA        st^`  1
L     REPEAT      of^`  1
L     DATA        ord   1
L     REPEAT      erw   1
L     DATA        ire   1
L     REPEAT      ^M^J@ 1
L     COMMAND     yF(   1
L     TIS   DOC   1
*     1
```

10. Bug List and Fixes for the Audio ALE Clean Tones Compact Disc 01a

******** Entries for ALE Clean Tones CD-0l (Beta test) ********

1. 9-21-93, Test 14A, D. Wortendyke, Incorrectly coded protocol, corrected for next release. The intent of Test 14A was to have a two-word net address, and a one word caller address, named IKE. Somehow GEORGE got into the file, and so the net slot time of 20 Tw's was too short. CD-01A will use IKE as the caller. Ignore this test on the Beta Disc.

2. 9-8-93, Test 10B, D. Wortendyke & G Teggatz, bogus test, changed test address of caller to DUM. The test plan calls for a two-word calling address and two-word caller address. This violates the length in the STD. The next release will use the caller address of DUM. It was found that some radios can decode the bogus call and respond, and some correctly ignore the call and immediately return to scan.

3. 3-20-94, Several Star NET & GROUP calls contained slot width errors and inconsistencies. These have been changed for the CD-01A, and are incorporated on the attached floppy disk.

******** Entries for ALE Clean Tones CD-Ola ********

1. 6-28-94, D. Wortendyke, Summary of changes in the CD-01A release. Since the Beta test CD in August 1993, the following tests have been revised or are new: 6E, F, G, H, J, 7A, B, C, D, E, F, 14A, 12A. Several other tests created by ITS on the Beta test CD, have been merged into companion or associated tests on this CD.

11. ALE Clean Tones On Compact Disc 01a Readme File
NTIA/ITS, Boulder, CO - January 1995

The files on this floppy disk and the zipped Internet package provide information for the proper use of the proposed audio Compact Disc (CD-01a). The purpose of Compact Disc 01a titled "ALE HF Radio Clean Tones for Interoperability Testing per FED-STD-1045A" is to provide an accurate, reliable, cost-effective method to test Adaptive HF radios for interoperability with Federal Standards 1045A and 1046/1. The MIL-STD-188-141A, Change #2 also contains these functions.

To get started, there are several files which must be printed. Each primary documentation file is provided in three forms. The file extension identifies the type of file. Text files end in ".TXT;" PostScript files end in ".PS;" and Word Perfect 5.x files end in ".WP5." You may choose which to use, depending upon your printing capabilities. The Word Perfect files use a PostScript printer. This file, README.1ST, and all the computer input/output files that have an extension of ".PRO" and ".TON" are only provided in ASCII text versions. A file extension of "xxx" implies that all three versions are provided on the floppy. ASCII text versions be printed in 12 chars/inch (elite).

FILES ON FLOPPY AND THE ZIPPED INTERNET PACKAGE:

1. FTIP-MAN.xxx and FTIP-AP1.xxx are the set of ALE test procedures derived from a Test Acceptance Plan (TAP) approved by the High Frequency Industries Associates (HFIA) during their meeting in San Diego, January 1993. Gene Teggatz, Rockwell, made a few corrections to the procedures to eliminate some problems encountered during early testing with a prototype CD by ITS in August 1993. The revised HFIA test document was received by ITS September 13, 1993. Only a few editorial changes have been made in the TAP to obtain the body of the FTIP-MAN.xxx document. The second portion of the TAP is Appendix I which describes the actual tests performed by the test tape or compact disc. ITS has modified the FTIP-AP1.xxx Appendix and tailored it to match the CD calls as of January 1995. The combined FTIP-MAN.xxx and FTIP-AP1.xxx document was approved by the FTSC HF as the test procedure for ALE radios with the ITS produced audio compact disc of ALE tones.

2. CDLAYOUT.xxx is the CD track listing. It also shows the correspondence between the CD track number and the test section number in Appendix I of the ALE Test Plan - Suggested Format of Standard Test Audio Compact Disc (Appendix C of this report.)

3. EVALUATE.xxx is the evaluation form you may use to provide user comments, evaluation, suggestions, and feedback about the CD and the concept for testing ALE HF radios. If preferred, you may write a letter on your letterhead with your evaluation of the CD. Although this is the final version of the files for the interoperability test disc, we welcome comments and suggestions. It will be assumed that persons or organizations that do not respond at all have not used or do not have a use for the CD, and may be dropped from future mailing lists.

4. *.PRO files found in the PRO subdirectory are the ALE protocol files from which the digital sound files were created. The problems found with the prototype Beta Test CD have been documented in both the FTIP-AP1 Appendix and the protocol files (*.pro). If you think there is a mistake in any of the protocols, we urge you to document your understanding of the correct interpretation of the call, and notify us by e-mail, FAX, or letter. The file names of the protocol files are ALE-nna.PRO, where nn = the section number found in Appendix I to the ALE Test Plan - Suggested Format of Standard Test Audio Compact Disc (Appendix C of this report.) The numbers range from 01 to 14 including the additional tests added by ITS. The letter "a" = the segment of the test, such as A, B, C, etc.. Many of the net and group calls have been modified since the Beta CD.

5. The four sample *.TON files found in the TON subdirectory are representative of the over-the-air encoded tones that are found on the audio CD. Each of the 8-ary tones is represented by an octal number (0-7) representing the gray-scaled ALE frequencies according to the table on page 15, Section 5.1.2, of FED-STD-1045. (0=750Hz, 1=1000Hz, 2=1500Hz, etc.) Each line of the *.TON files has 49 tones, or exactly one ALE word of 392 ms. The file names correspond exactly with the ALE-nna.* names in the PRO subdirectory. The two files that represent ALE calls are for tests 5A and 6A. The other two are calibration tone files. They may have any length line and were chosen to have 50 tones equal to 0.4 seconds. They are named ALE-00B.TON and ALE-00C.TON.

6. D-INFO.xxx file is a set of documentation explaining how the audio disc was built from the FED-STD-1045A and FED-STD 1046/1 protocols and addresses. Additional information about the CD and future plans may be found here. This is a very important file that includes equipment set-up for running the tests on the CD.

7. *.EXE, executable binary programs have been added. Five programs are included on this disk: ALECALL.EXE which generates the sound files from the *.PRO protocol files, ALETIME.EXE which computes the playing time of the protocol files, and three other utility programs that build protocol files for the user. They are: C-AMD.EXE to build calls with AMD messages, C-DTM.EXE for DTM messages, and C-LQA.EXE to add LQA exchange to a call. These are in the subdirectory "exe" on the floppy disk. The ALECALL software will generate only clean tones, ignoring commands in the protocol files to generate noise, fading, and multipath.

Authorship, sponsor, and additional information:
The work to produce audio Compact Disc 01a was performed by:

Wireless Networks Group 303-497-5116 (Group Secretary)
US Department of Commerce 303-497-6982 (FAX)
NTIA/ITS.N1 tsparkman@its.bldrdoc.gov (E-mail)
325 Broadway
Boulder, CO 80303-3328

Additional information about the research, development, and applications of the ALE Compact Disc may be found in:

QEX (ARRL Experimenter's Exchange), "The Growing Family of Federal Standards for HF Radio ALE, Part II: A Compact Disc for Testing HF ALE Radios," by D. Wortendyke, C. Riddle, and D. Bodson, pp. 9 - 14, August 1993.

E-Mail questions, evaluations, criticism, and suggestions should be sent to:
ale-cd@its.bldrdoc.gov or FAXed to "ALE-CD" 303-497-6982

************ WARNING ************

Information pertaining to use of Compact Disc 01a should be directed to NTIA/ITS or NCS at the above addresses. The CD represents a working implementation of a Federal Standard, and adheres to the 10 ppm accuracy required by FED-STD-1045. The civilian and military standards upon which this audio Compact Disc was based are under the control of the U.S. Government and its authorized Agencies. The Federal Government reserves the right to change the audio disc or standard at any time, and will notify all known recipients of the current compact disc of the availability of the newly revised audio compact disc.

The ALE Clean Tone audio compact disc and the tone files that make up the collection of audio sounds used for testing the FED-STD-1045A and FED-STD-1046/1 HF radios are copyrighted (c) 1994. This collection of tones, or audio information, is referred to as NIST Special Database 17. The tone files are not provided on the floppy so that the disk may be freely copied. You may generate the tone files for all the CD using the software and protocol files provided but they are for local use only. The sound files and the tone files make up the copyrighted database.

The ALE Clean Tone Database is maintained by NTIA/ITS in Boulder, Colorado. Specifically the Wireless Networks Group (ITS.N1) developed the Special Database of ALE tones under the sponsorship of NCS, Office of Technology and Standards, (c) 1994 copyright by the U.S. Department of Commerce on behalf of the United States. All rights reserved. No part of this database may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the distributor. This copyright is assigned to the Standard Reference Data Program, National Institute of Standards and Technology (NIST).

12. Calculations for Slot Widths on Compact Disc 01a

Star Net Calls (06), Revised: 6-16-94

Test#	**Slot#**	**Caller, # of words- Trw**	**Responder, #of words-Trw ***	**LQA (or AMD)**	**Guard Tw**	**Values 3(2*C+R+L)+5**	**Width-Tw**
06A	0	"IKE" = 1	R=C=1	0	5	3*(2*1+1+0)+5	14
	1	1	"ABC" = 1	0	5	ditto	14
	etc	1	1	0	5	ditto	14
06B	0	"IKE" = 1	R=C=1	1	5	3*(2*1+1+1)+5	17
	1	1	"ABC" = 1	1	5	ditto	17
	etc	1	1	1	5	ditto	17
06C	0	2	R=C=2	1	5	3*(2*2+2+1)+5	26
	1	2	2	1	5	ditto	26
	etc				5		
06D	same	as test	#06C		5		26
06E	same	as test	#06A		5		14
06F	same	as test	#06C		5		26
06G	same	as test	#06A		5		14

* For SLOT 0, set R = C

Star GroupCalls (07), Revised: 6-16-94

Test#	**Slot#**	**Caller, # of words- Trw**	**Responder, #of words-Trw ***	**LQA (or AMD)**	**Guard Tw**	**Values 3(2*C+R+L)+5**	**Width. Tw**
07A	0	"RONNIE"=2	R = C = 2	0	5	3*(2*2+2+0)+5	23
	1	2	"ABC" = 1	0	5	3*(2*2+1+0)+5	20
	2	2	"DEF" = 1	0	5	ditto	20
	3	2	"GHIJ" = 2	0	5	3*(2+2+2+0)+5	23
	4	2	"456789DEF" = 3	0	5	3*(2*2+3+0)+5	26
	5	2	5	0	5	3*(2*2+5+0)+5	32
07B	0	2	2	1	5	3*(2*2+2+1)+5	26
	1	2	"ABC" = 1	1	5	3*(2*2+ 1 +0)+5	23
	2	2	"DEF" = 1	1	5	ditto	23
	3	2	2	1	5	3*(2*2+1+1)+5	26
	4	2	3	1	5	3*(2*2+2+1)+5	29
	5	2	5	1	5	3*(2*2+5+1)+5	35
07C	same	as test	#07B		5		
07D	0	"RON"= 1	R = C = l	0	5	3*(2*1+1+0)+5	14
	1	1	"ABC" = 1	0	5	ditto	14
	2	1	"DEF" = 1	0	5	ditto	14
	3	1	"GHIJ"= 2	0	5	3*(2*1+2+0)+5	17
07E	0	"RON"= 1	R=C=l	0	5	3*(2*1+1+0)+5	14
	1	1	"ABC"= 1	0	5	ditto	14
	2	1	Null slot	0	5	ditto	14
	3	1	"DEF"= 1	0	5	ditto	14
	4	1	"GHIJ"= 2	0	5	3*(2*1+2+0)+5	17

* For SLOT 0, set R = C